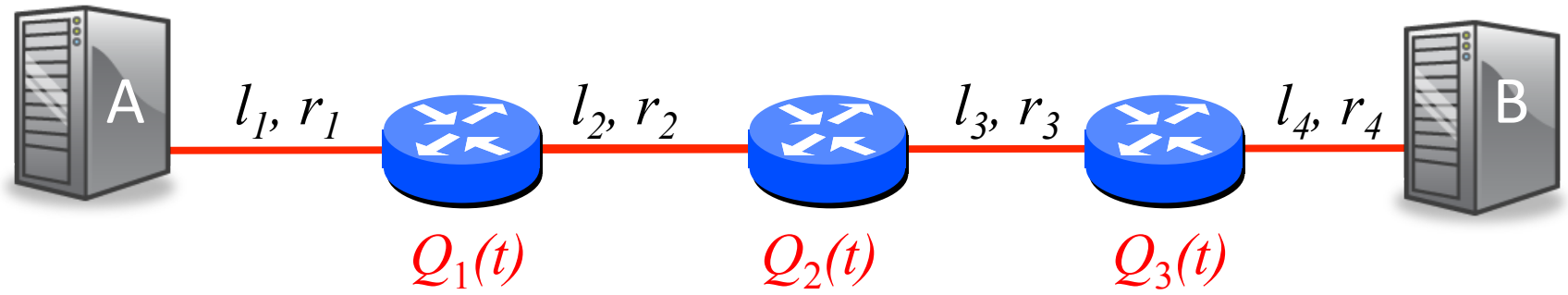
CS144
An Introduction to Computer Networks

Packet Switching

Guaranteed Delay

Nick McKeown
Professor of Electrical Engineering
and Computer Science, Stanford University

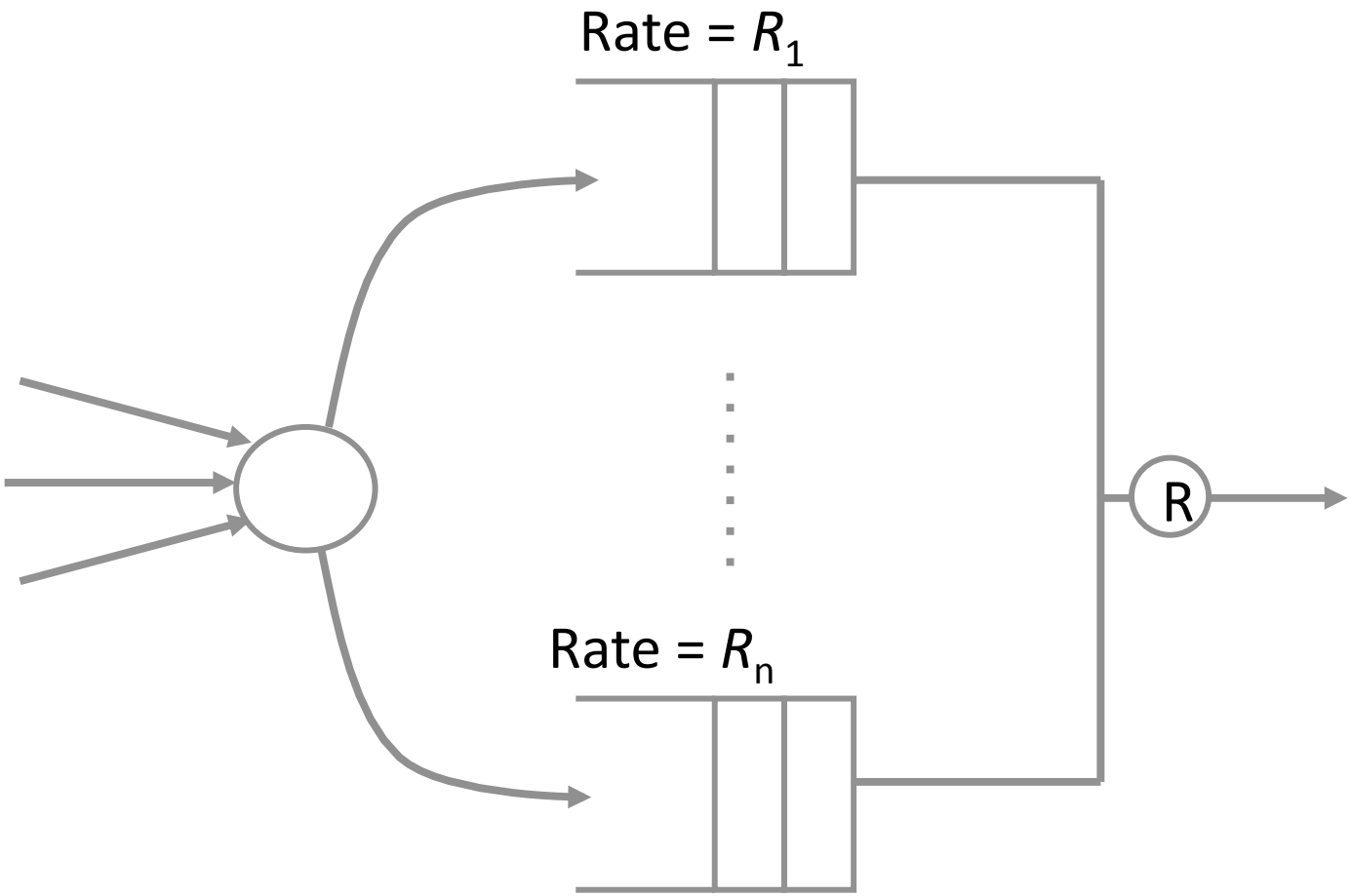
Delay guarantees: Intuition



$$\text{End-to-end delay, } \tau = \sum_i \left(\frac{p}{r_i} + \frac{l_i}{c} + Q_i(t) \right)$$

If we know the upper bound of $Q_1(t)$, $Q_2(t)$ and $Q_3(t)$, then we know the upper bound of the end-to-end delay.

Delay guarantees: Intuition



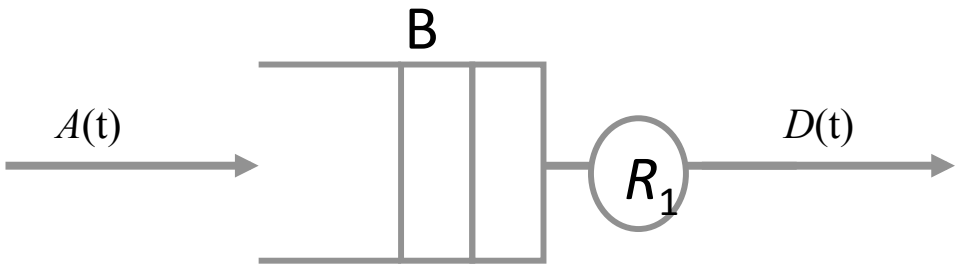
So how can we control the delay of packets?

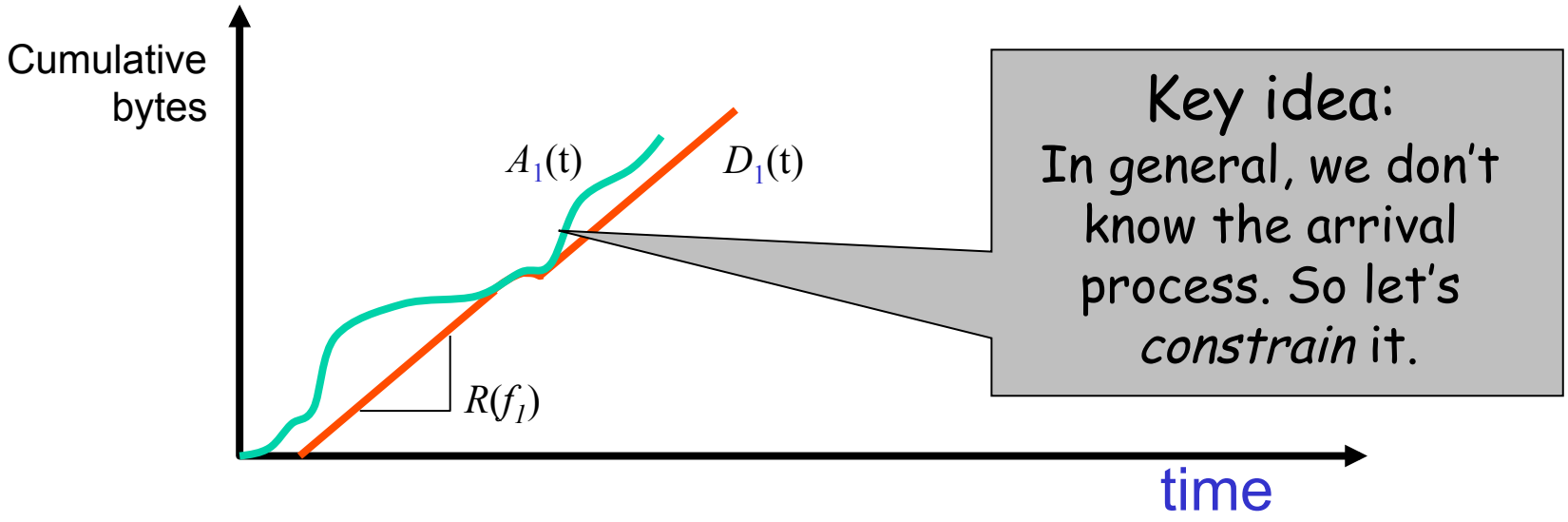
What we already know how to control:

1. The rate at which a queue is served (WFQ).
2. The size of each queue.

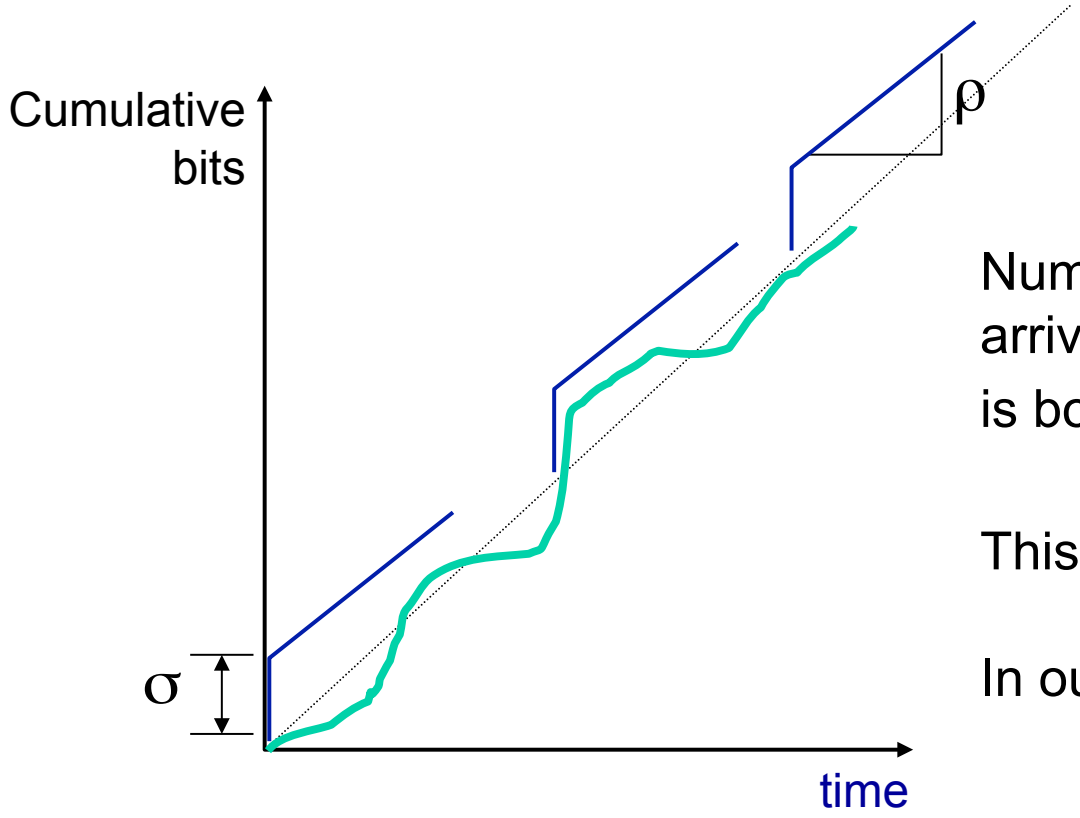
How do we make sure no packets are dropped?

Zooming in on one queue




Cumulative bytes
$A_1(t)$
$D_1(t)$
$R(f_1)$
time
Key idea:
In general, we don't know the arrival process. So let's *constrain* it.

Constraining traffic

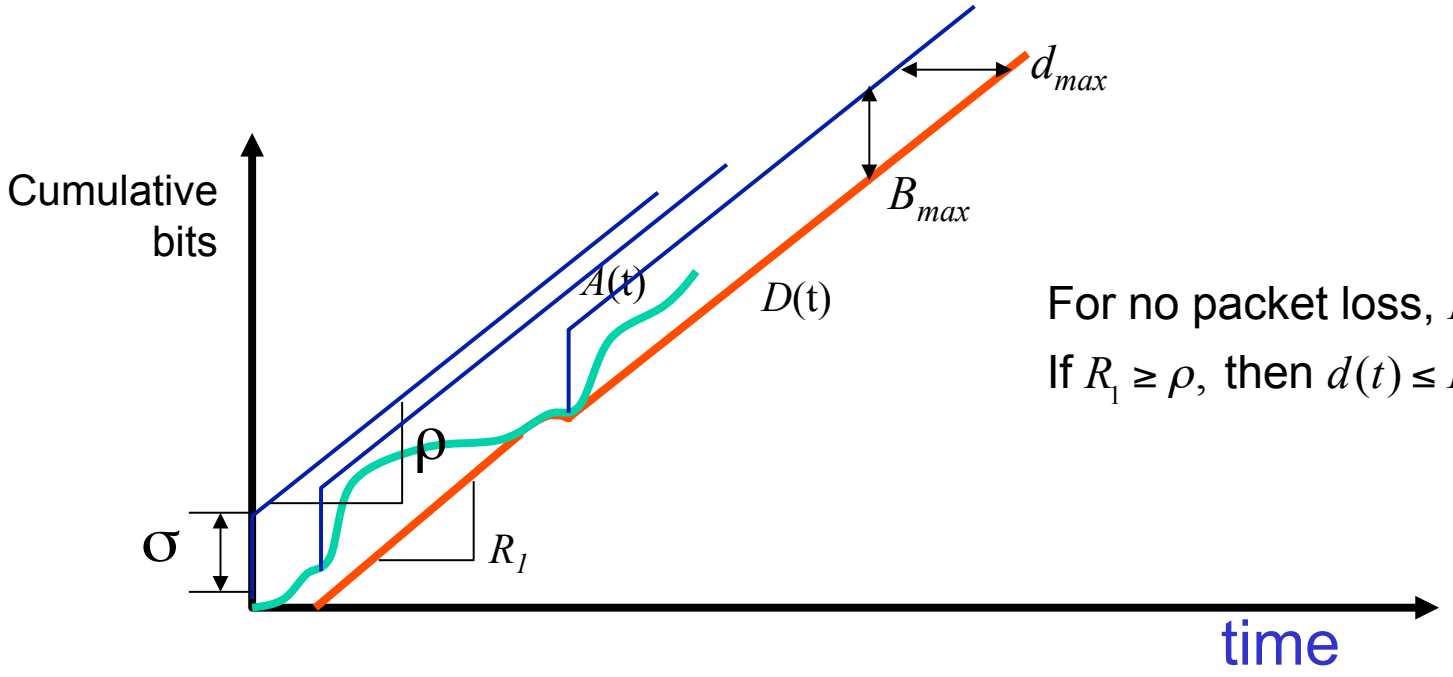


Number of bits that can arrive in any period of length t is bounded by: $\sigma + \rho t$

This is called "(σ,ρ) regulation"

In our example: $\sigma = B$ and $\rho = R_1$

(σ,ρ)-constrained Arrivals and Minimum Service Rate

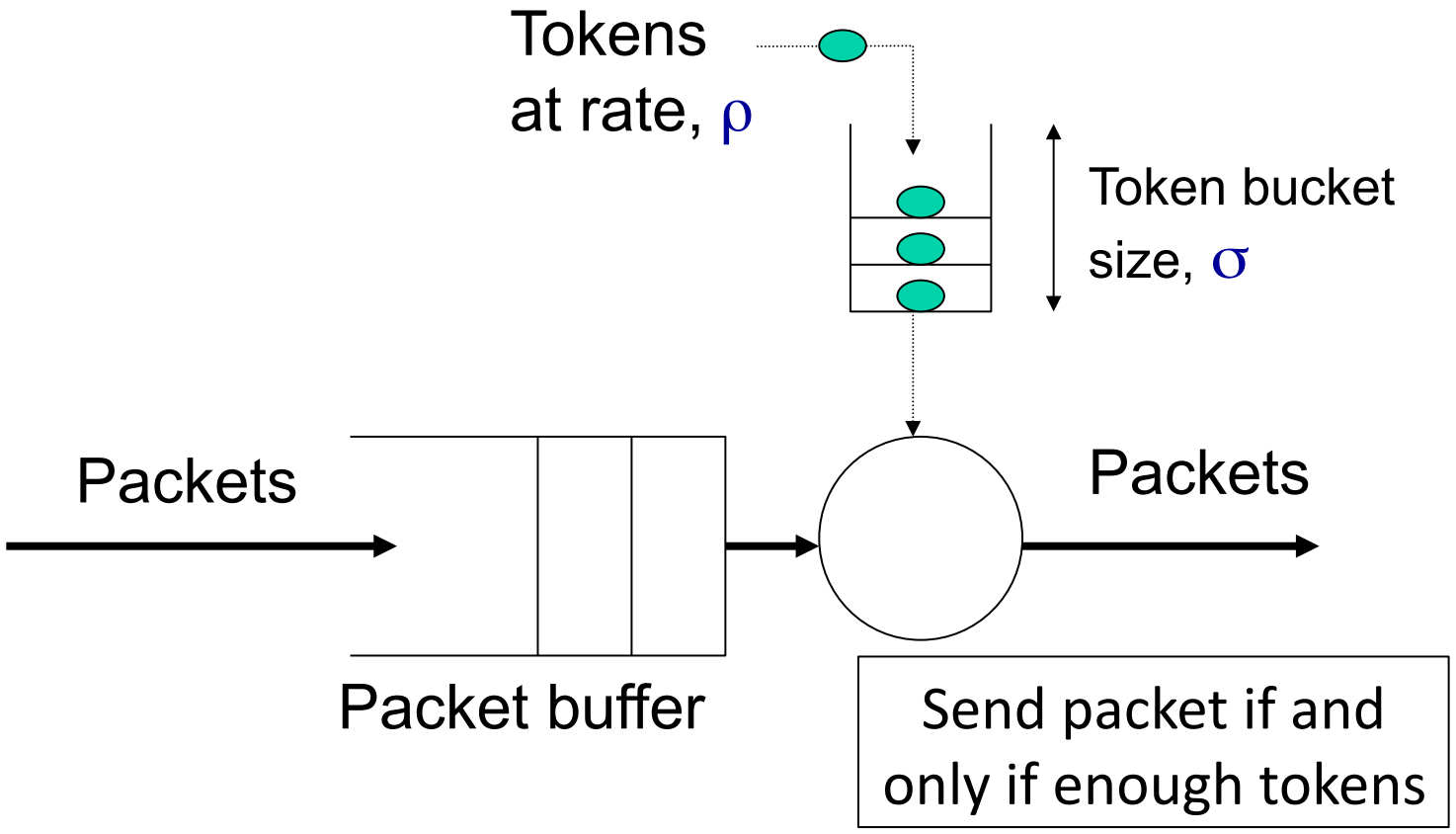


For no packet loss, $B \geq \sigma$.

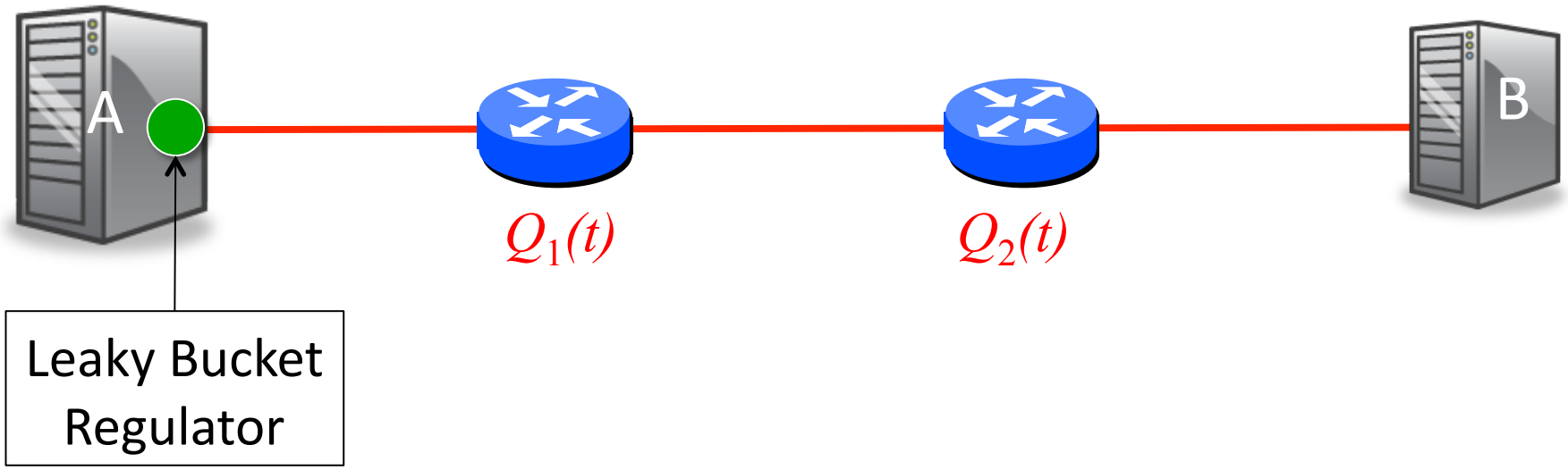
If $R_1 \geq \rho$, then $d(t) \leq B / R_1$.

If flows are leaky-bucket constrained, and routers use WFQ, then end-to-end delay guarantees are possible.

The leaky bucket regulator

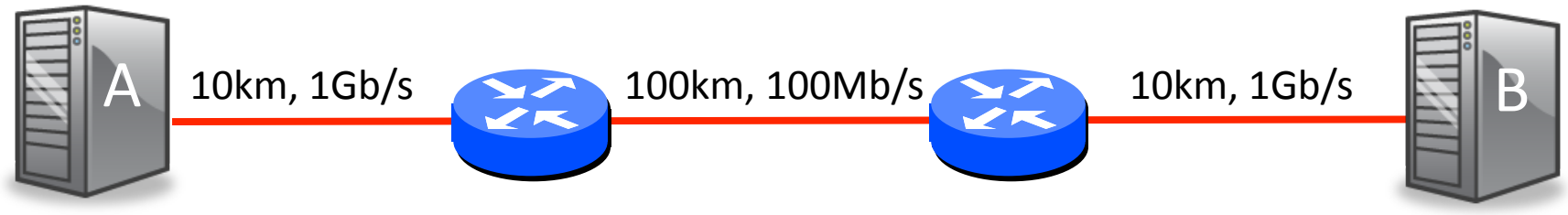


Putting it all together



An example

In the network below, an application wants a rate of 10Mb/s and an end to end delay of less than 5ms for 1000byte packets.



Once we decided to bound the delay to no more than 2.15ms in each router, we concluded that we need to make sure: (a) we don't store more than 2960 bytes in each router, which we accomplish by setting the token bucket at the source equal to 2960 bytes, and (b) the router buffer can hold at least 2960 bytes so we don't drop data (we can make the router buffer bigger if we'd like to; we just won't use it).

In practice

While it is technically possible to do so, very few networks actually control end to end delay.

Why?

- It is complicated to make work, requiring coordination.
- In most networks, a combination of over-provisioning and priorities work well enough.

Summary

If we know the size of a queue and the rate at which it is served, then we can bound the delay through it.

We can pick the size of the queue, and WFQ lets us pick the rate at which it is served.

Therefore, we just need a way to prevent packets being dropped along the way. For this, we use a leaky bucket regulator.

We can therefore bound the end to end delay.